

Le système de fichiers

par [Romain Guy \(Gfx\)](#)

Date de publication : 6 Juin 2006

Dernière mise à jour :

Un système Linux gère les fichiers comme une arborescence structurée de manière hiérarchique. Mais ceci ne nous éclaire pas sur la façon dont sont stockés les fichiers et répertoires sur le disque dur.

- I - Introduction
- II - Définition d'un système de fichiers
- III - Structure physique
- IV - Constitution d'un système de fichiers

I - Introduction

L'[article précédent](#) portait sur la compilation du noyau. Si vous avez tenté l'aventure, la question des types de systèmes de fichiers à intégrer au noyau vous a sûrement quelque peu interloqué. Linux connaît en effet plusieurs types de système de fichiers tels que l'Extended File System version 2 ou le Reiser File System. Mais avant d'étudier plus avant la gestion interne des fichiers sous Linux, il convient de faire la distinction entre une arborescence et un système de fichiers.

II - Définition d'un système de fichiers

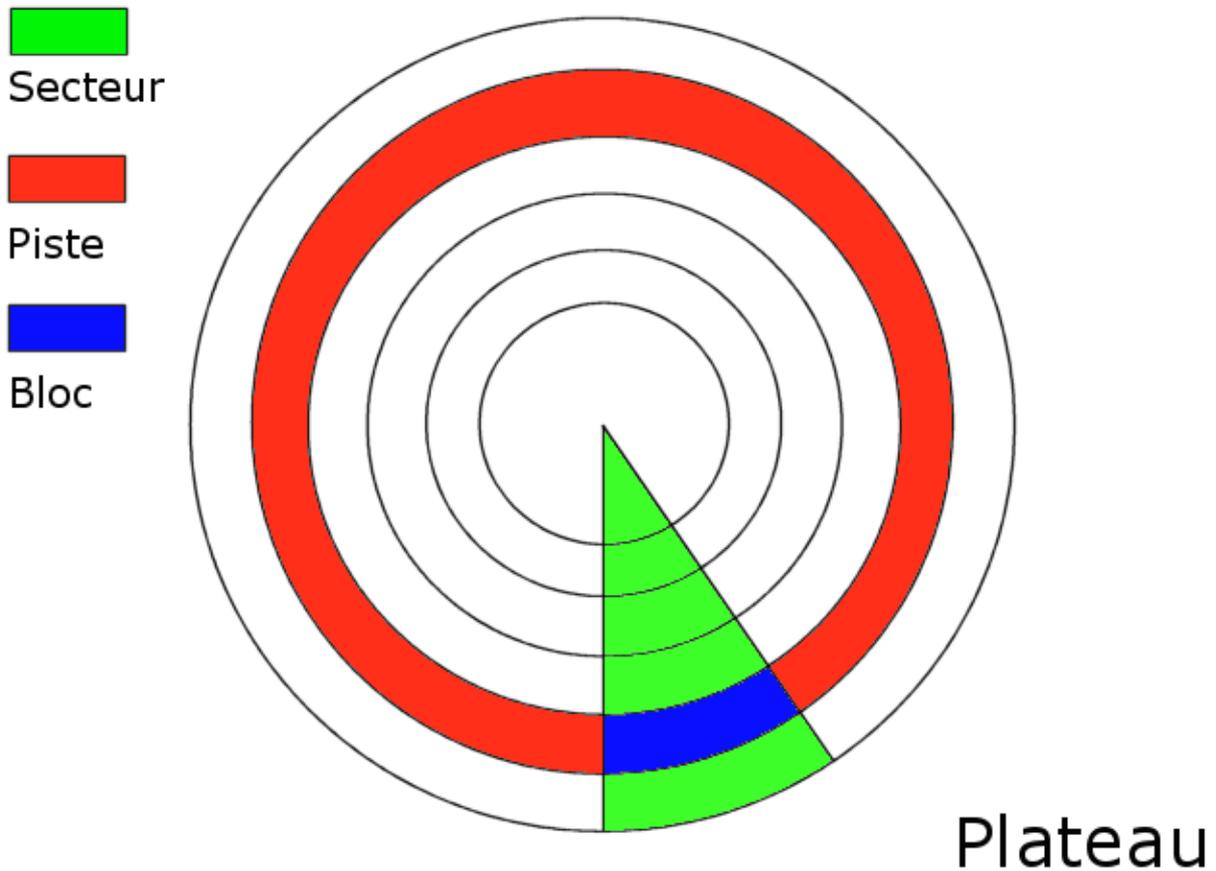
La notion d'arborescence désigne la représentation logique des fichiers et des répertoires telle qu'elle apparaît au final pour l'utilisateur du système. Une telle représentation est incarnée par un arbre, donc par une succession de noeuds ramifiés. Chaque noeud peut représenter un répertoire ou un fichier. La notion de répertoire va bien au-delà de celle liée au système Windows par exemple. En effet, un noeud peut contenir tout un système de fichiers ! Ainsi un support amovible tel qu'un CD-Rom ou une disquette se voit présent dans l'arborescence, usuellement dans le répertoire /mnt. Ainsi, dans une distribution Red Hat, le répertoire /mnt/cdrom contient en réalité non pas des données présentes sur le disque dur, mais un système de fichiers de type CD-Rom (iso9660, Joliet...). Un système de fichiers est donc une représentation logique et structurée des données physiques d'un média. Les systèmes de fichiers dévoilent alors toute leur puissance: un média amovible, un ordinateur distant, toute source de données peut être intégrée au sein de l'arborescence. On obtient au final une impression d'homogénéité des données accessibles. Voici une liste non exhaustive des systèmes de fichiers existant sous Linux :

- ext2fs: Le système de fichiers le plus employé sous Linux. Il est le successeur de l'extfs.
- isofs: Le ISO FS est le système de fichiers pour les CD-Rom à la norme ISO-9660.
- minixfs: Le système de fichiers de l'ancêtre de Linux, Minix.
- msdosfs: Offre un accès aux lecteurs DOS.
- nfs: Système utilisé par tous les Unix pour intégrer à l'arborescence un ordinateur distant.
- proc: Pseudo-système de fichiers donnant des informations sur le noyau.
- vfat: Système de fichiers de type VFAT (Windows 95...)

III - Structure physique

La description exacte d'un système de fichiers dépend bien évidemment du support utilisé. Un système de fichiers comme le NFS (Network File System) ne fonctionnera évidemment pas comme le ext2fs standard. Nous allons nous concentrer uniquement sur la structure physique d'un système de fichiers représentant un disque dur ou une partie de disque dur. L'espace disponible sur un disque dur est exprimé en blocs. Chaque bloc a une taille de 512 octets. Chacun de ces blocs possède un numéro et le contrôleur du disque dur s'arrange pour les numéroter de manière continue. Cette numérotation permet d'obtenir la position d'un bloc sur le support en fonction de son numéro. Ainsi, le bloc numéroté 2048 sera le 2048-ième sur le disque. A partir de cette information, le contrôleur est en mesure de situer le bloc exactement à la surface du disque en jouant sur la division physique du disque. Un disque dur est en effet constitué d'une " pile " de plateaux. Or, chacun de ces plateaux est divisé en secteurs, sortes de " tranches " partant du moyeu, et en piste. Les intersections des pistes et des secteurs forment les fameux blocs (voir figure 1).

Hélas, tout cela serait bien trop simple si tous les systèmes de fichiers possédaient le même découpage des blocs physiques en blocs dits logiques. Chaque type de système regroupe plusieurs blocs physiques en un seul logique. Le seul point commun entre tous les systèmes de fichiers correspond au nombre de blocs physiques utilisés qui eux possèdent la taille fixe de 512 octets. Ainsi, la dénomination " bloc " correspondra dans la suite de cet article à un bloc logique.



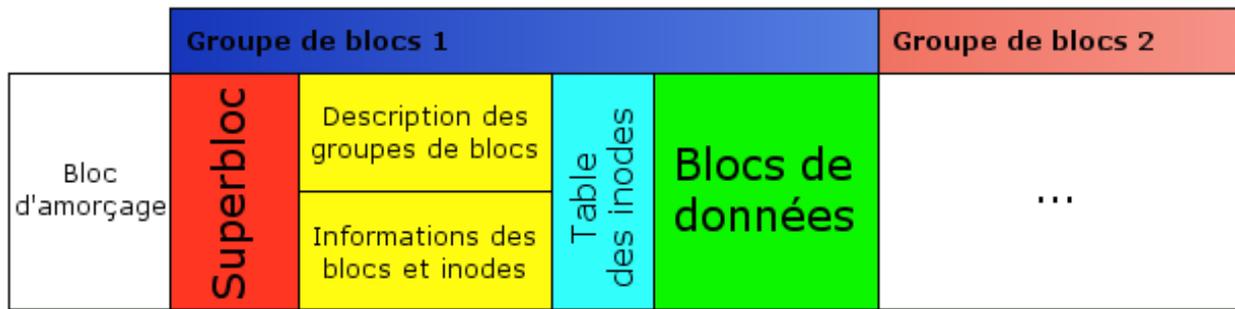
Structure physique d'un disque dur.

IV - Constitution d'un système de fichiers

Eu égard la grande diversité des systèmes de fichiers disponibles, il y a peu de chances qu'un système possèdent beaucoup de points communs avec un autre. Nous resserrerons donc notre étude sur le type ext2fs exclusivement. Ce dernier possède les caractéristiques suivantes :

- un bloc d'amorçage
- un superbloc
- une table des inodes
- des blocs de données

Le bloc d'amorçage est le premier, numéroté 0, d'un système de fichiers. Il peut contenir un programme chargé d'initialiser le système. Normalement, seul le premier système de fichiers, si plusieurs partitions existent, possède un bloc d'amorçage non vide. Dans le cas du système ext2fs, la taille de ce bloc est de 1024 octets, soit un kilo octet. A la suite du bloc d'amorçage, ce trouvent des groupes de blocs. Chaque groupe contient un superbloc, suivi d'une description des groupes de blocs, puis d'informations sur les blocs ou inodes occupés et enfin viennent la table des inodes et les blocs de données (voir figure 2). Nombreux sont ceux qui doivent connaître la signification de " inode ". Sous Linux, un inode est destiné à gérer les informations d'un objet du système de fichiers. Par exemple, ce sont les inodes qui gèrent les droits d'accès ou les liens symboliques.



Répartition des blocs logiques.

Comme nous venons de le voir, un groupe de blocs contient six éléments (les informations des blocs et inodes comptant pour deux sur la figure ci-dessus). Le système ext2fs est donc constitué d'une multitude de groupes de blocs dont la taille est de 8 192 blocs dont 2 048 inodes. Parmi ces derniers, les 8 premiers sont réservés par le système. Il en reste donc 2 040 à disposition de l'utilisateur.

Le superbloc contient toutes les informations concernant le système de fichier. En cas d'erreur sur un superbloc, il est possible de récupérer ces informations sur un autre. Voici les informations les plus importantes que l'on peut trouver au sein d'un superbloc:

- taille totale du système de fichiers, en blocs et inodes
- nombre de blocs libres
- nombre de blocs réservés aux inodes
- taille d'un bloc de données
- heure de la dernière modification effectuée
- heure de dernière vérification du système de fichiers

La liste de description des groupes de blocs facilite l'accès aux blocs de données. Cette liste définit les blocs physiques contenant les blocs de données, les inodes. Elle répertorie également le nombre de blocs ou d'inodes libres.

Les informations des blocs et des inodes est une image binaire de l'occupation des blocs et inodes. Codées sur 1024 octets, ces informations indiquent un bit mis à 0 pour un bloc libre ou à 1 pour un bloc utilisé. Avec ces 1024 octets, il est possible de se renseigner sur l'état de 8 192 blocs ou inodes. On retrouve ici la limite d'un groupe de blocs.

La table des inodes répertorie les 2 048 inodes les uns à la suite des autres. Un inode est essentiel pour permettre l'accès aux fichiers car il décrit toutes les caractéristiques d'un fichier donné. Pour chaque fichier enregistré sur le disque dur, un inode est associé. Ainsi un inode peut stocker les informations suivantes:

- identification du propriétaire
- taille en octets
- heures de modification/création/accès
- nombre de liens liés à ce fichier
- type de fichier: classique, répertoire, lien symbolique...
- droits d'accès

Enfin, les blocs de données occupent l'espace compris entre la fin de la table des inodes jusqu'à la fin du groupes de blocs. Chaque bloc de données a exactement la taille d'un bloc logique. Vous l'aurez compris, ce sont les blocs de données eux mêmes qui renferment les données de votre ordinateur. Les accès aux blocs de données sont possibles grâce à la table d'adressage des blocs de données. Cette table contient en tout et pour tout 15 enregistrements. Chacun d'entre eux renvoie à un bloc de données. La structure des ces enregistrements permet d'adresser un total de 16,6 Go, ce qui est la limite d'une partition ext2fs. Outre cette limitation, le fait que la référence au prochain caractère à lire ou à écrire soit indiquée par un entier non signé de 32 bits limite la taille d'un fichier à 2^{32} , soit 4 294 967 295 octets. C'est à dire 4 Go.